

분산 멀티미디어 환경에서 실행되는 결함 허용 시스템의 성능 분석

고 응 남

천안대학교 정보통신학부

요 약

멀티미디어는 현재 다양한 분야에서 응용되고 있다. 특히, 멀티미디어 교육 시스템을 위한 공동 작업 환경에 대한 관심도 증가하고 있다. 두레는 분산 멀티미디어 환경과 멀티미디어 원격 교육 시스템에서 실행되는 응용들의 개발을 지원하기 위한 프레임워크이다. EDA는 분산 멀티미디어에서 하나의 소프트웨어 오류를 자동적으로 감지할 수 있는 시스템이다. 이 시스템은 효율적인 두레 환경의 구성과 실험을 위하여 구성되고 구현되었다. 세션과 관련된 프로세스를 정기적으로 폴링하여 오류를 감지한다. 기존 방식은 모든 프로세스들을 정기적으로 폴링하여 오류를 감지한다. 본 논문은 규칙-기반 SES와 DEVS 모델링과 시뮬레이션 기법을 사용하면서 분산 멀티미디어 상에서의 오류 감지 시스템의 성능 분석을 설명한다. DEVS에서 하나의 시스템은 시간, 입력, 상태, 출력 및 함수들을 가지고 있다.

Performance Analysis of Fault Tolerance System on Distributed Multimedia Environment

Eung-Nam Ko

ABSTRACT

Multimedia is now applied to various real worlds. In particular, the focus of CSCW(Computer Supported Cooperated Work) for multimedia education system has increased. DOORAE is a framework for supporting development of applications running on distributed multimedia environment and multimedia distance education system. EDA is a system is able to detect automatically a software error based on distributed multimedia. It has been designed and implemented for construction and experiment of effective DOORAE environment. It detects an error by polling periodically the process with relation to session. Conventional method detects an error by polling periodically all the process. This paper explains a performance analysis of an error detection system running on distributed multimedia environment using the rule-based SES and DEVS modeling and simulation techniques. In DEVS, a system has a time base, inputs, states, outputs, and functions.

I. 서론

멀티미디어는 교육, 원격 진료 등 다양하고 광범위한 분야에서 응용되고 있다. 멀티미디어 이용 분야 중에서 교육 분야에서도 많은 관심을 가지고 있다[1,2,3,4]. 사회가 복잡해지고 컴퓨터 네트워크가 발달함에 따라 다양한 종류의 상호 참여가 요구되어지고 있다. 이에 따라 상호 참여를 제공하기 위한 화상 회의 시스템의 개발이 활발해지고 있다. 화상 회의 시스템과 공동작업(CSCW: Computer Supported Cooperative Work) 환경은 원격지 전문가로부터 여러 장소에서 서로의 모습을 보며 음성과 연필로 서로의 의견을 교환할 수 있게 하였다. 분산 멀티미디어 정보 시스템에 대한 요구는 매우 빠르게 상업, 생산, 교육, CAD(Computer Aided Design)/CAE(Computer Aided Engineering), 의학, 기상 등 많은 분야에서 필요로 하고 있다[5,6]. 최근 들어 이러한 멀티미디어 교육 시스템의 공동 작업 환경이 증가하고 있는데 반하여 이러한 시스템에서의 전체적인 망 관리, 특히 응용 S/W에 대한 결함을 발견 및 복구하는 연구는 미흡한 실정이다. 분산 시스템은 하나의 노드 또는 자원에 결함이 발생해도 전체 시스템에는 큰 손실을 입히지 않는 특성을 가지기 때문에 결함 허용 시스템의 설계를 위한 좋은 조건을 가진다. 시스템에서 결함을 허용하는 정도에 따라 결함 허용 시스템은 소프트웨어 기법, 하드웨어 기법, 혼합 기법 등으로 분류할 수 있다[7,15]. 인터넷 보급이 급격하게 증가함에 따라 인터넷을 활용한 다양한 솔루션들이 소개되고 있다. 최근 들어 초고속 통신망 구축이 현실화되면서 멀티미디어를 포함하는 콘텐츠가 더욱 다양화되어 가고 있다. 그 예로서, 인터넷을 활용한 전자상거래가 생활의 한 부

분으로 사용되고 있으며, 교육에서도 웹기반교육, 실시간/비실시간 원격교육과 같은 콘텐츠가 개발되고 있는 것이다. 따라서, 본 논문에서는 분산 멀티미디어 환경에서 응용 S/W의 결함을 미리 감지하여 알려주고 복구할 수 있는 에이전트 중에서 오류를 감지하는 에이전트를 DEVS(Discrete Event Specification) 형식론(formalism)의 계층적 구조를 이용하여 효율성을 분석, 비교한다. 특히, 결함을 감지하는 방식은 여러가지가 있는데 그 중에서 폴링(polling) 방식에 대해서 범위를 좁히려고 한다. 폴링 방식은 시스템에 대한 부하가 걸린다는 단점을 갖고 있다. 본 논문에서는 전체 응용에 대해서 폴링 방식을 사용하는 방식과 폴링을 사용하지 기존 시스템의 정보를 갖고 있는 전체 세션 관리자를 이용하여 등록되어 있는 필요한 응용만 찾아서 폴링 시간을 줄이는 방식 2가지를 비교하여 효율성 검토를 할 수 있는 시스템의 모델링을 제안한다. 본 논문의 구성은 2에서 관련 연구를 기술하고, 3에서는 제안하는 분산 멀티미디어 환경에서의 EDA를 기술하고, 4에서는 오류 감지 시스템의 모델링, 5에서는 결론을 기술한다.

II. 관련 연구

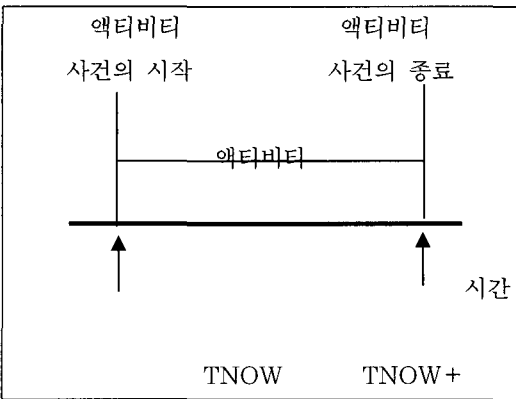
이 장에서는 이산 사건 모델링(Discrete Event Modeling)과 DEVS의 형식론에 대해서 기술한다.

2.1 이산 사건 모델링

이산 사건 모델링(Discrete Event Modeling)의 목적은 엔티티(entity) 들이 포함된 액티비티(activity)들을 재생하여 시스템의 잠재적 행태와 성능에 대한 것들을 연구하고자 하는 것이다. 이러한 목적은 시스템의 상태들을 정의하고, 시스템

을 어떤 상태에서 다른 상태로 옮기는 액티비티들을 구축함으로써 달성된다. 시스템의 상태는 엔티티들의 속성에 할당된 수치의 측면에서 정의가 된다. 모든 엔티티들이 어떤 상태에 대하여 정의된 속성값의 범위에 있을 때 시스템은 그 상태에 있다고 말한다. 따라서 이산사건 시물레이션은 시간에 따른 시스템 상태의 동적인 묘사이다[8].

이산 사건 모델링에서 사건과 액티비티간의 관계는 그림 1과 같으며, 액티비티가 시작하거나 끝나는 결정이 이루어지는 시점에 사건이 발생하고 있다. 프로세스(process)는 시간에 따라 정렬된 사건들의 순서이며, 여러 액티비티 들을 포함할 수 있다[8].



(그림 1) 사건과 액티비티 간의 관계

2.2 DEVS의 형식론

DEVS(Discrete Event System Specification)는 Bernard P. Zeigler에 의해 개발된 이산 사건 모델들의 계층 구조적 모듈화 방법을 제공해주는 형식론이다. 시스템을 작은 모듈들로 나누고 그것들로 전체 시스템을 계층적으로 구성해 나간다. 각 모듈들은 원자(atomic) 모델로 표현되며 그것들의

계층적 구성은 커플(coupled) 모델로 표현된다. 모델링 된 시스템의 시물레이션을 위해 추상화 시물레이터(Abstract simulator) 알고리즘이 제공된다. 추상화 시물레이터의 종류에는 시물레이터(simulator)와 협동자(coordinator)가 있으며 이것들은 각각 원자모델과 커플모델을 위한 시물레이터이다. 모델들과 추상화 시물레이터들은 일대일 대응 관계를 가진다. 즉, 하나의 모델은 하나의 추상화 시물레이터와 항상 쌍을 이루게 된다. DEVS 형식론에서 가장 기본이 되는 모델인 원자 모델은 다음과 같은 집합으로 표현된다[9-13].

- M = < X, S, Y, int, ext, λ, ta >
- X : 외부 입력 사건들의 집합
- S : 상태 변수들의 집합
- Y : 외부 출력 사건들의 집합
- int : 내부적 상태 변환 함수
- ext : 외부적 상태 변환 함수
- λ : 출력 함수
- ta : 시간 진행 함수

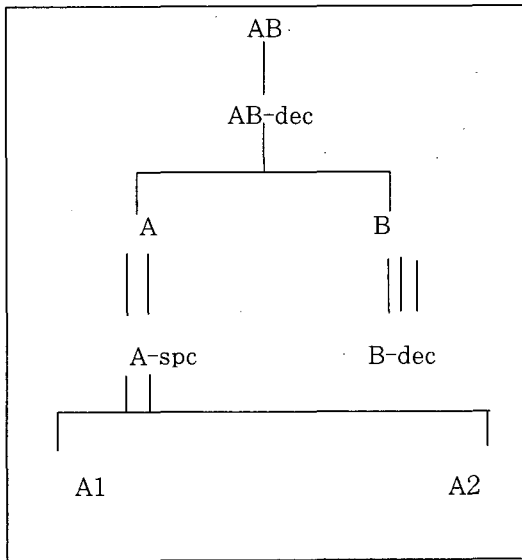
원자모델을 결합하여 새로운 결합 모델을 형성한다. 이 결합 모델은 또한 다른 모델의 구성 요소 모델이 될 수 있기 때문에 이것을 이용하여 복잡한 모델을 계층적으로 구성할 수 있게 된다. 결합 모델의 구조 표현은 다음과 같다[9-13].

- DN = < D, {Mi}, {Ii}, {Zij}, select >
- DN : Diagraph Network
- D : 구성 요소 모델들의 이름의 집합
- {Mi} : i번째 구성 요소를 이루는 기본 모델들
- {Ii} : i번째 모델의 influencees 모델들의 집합
- {Zij} : i번째 모델의 출력을 j번째 모델의 입력으로 연결하는 함수

select : 여러 구성 요소들이 같은 시간에 스케줄을 원할 때 그 중에서 하나를 선택하는 함수

2.3 SES의 형식론

Zeigler에 의해 제안된 SES(System Entity Structure)는 구성원들의 분할 (decomposition), 분류(taxonomy), 결합관계(coupling relationships), 제약조건(constraint) 등을 표현할 수 있는 구조체로서, entity, aspect, 그리고 specification이라는 3가지 모드로 구성되어 있다. 간단한 SES의 예가 그림 2에 나타나 있다[9,12,13].



(그림 2) 간단한 SES의 예

여기서, entity는 임의의 실체를 의미하며 여러 개의 aspect와 specification을 가질 수 있다.

Aspect는 entity의 분할 구조 관계를 나타낸다 즉, 그림 2의 한 줄의 수직선에 의해 표현된다. Specialization은 entity 종류 들의 분류 구조 관계

를 나타낸다(그림 2의 두 줄의 수직선에 의해 표현). Multiple entity는 동종의 entity들의 집합 관계를 나타내는 것으로, 시스템에서 갯수가 가변적인 여러 개의 entity를 표현할 때 사용된다 즉, 그림 2의 수직선 세 개에 의해 표현된다. 이러한 관계 표현들을 적절히 이용하면, 설계 대상 시스템의 모든 가능한 구조적 지식의 표현이 가능한 데, SES에 pruning이라는 선택 과정을 top-down으로 적용시키면 하나의 설계 구조가 구성될 수 있다 [13,14].

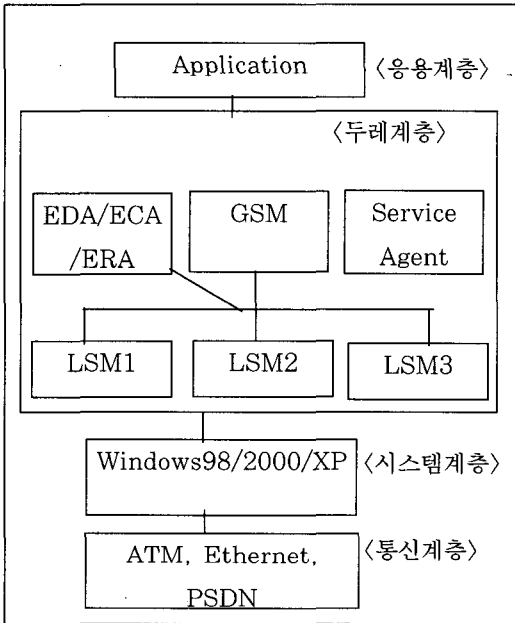
III. 분산 멀티미디어 환경에서의 EDA

본 논문에서는 멀티 에이전트의 특성인 각 에이전트의 자치권을 보장하면서 서로간의 협력을 통해서 전체적인 망 관리, 특히 응용 S/W에 대한 결함을 발견하고 복구하는 에이전트 중에서 오류를 빨리 발견하는 에이전트인 EDA(Error Detection based on Agent)를 실제계의 환경으로 모델링 한다.

3.1 분산 멀티미디어 환경

두레는 상호 참여형 멀티미디어 응용 개발 환경으로 그림 3처럼 4계층으로 구성되어 있다. 기존의 단일 멀티미디어 응용 개발에 발생하는 미디어 제어와 세션 관리에 대한 개발 과정의 비용을 줄임으로 상호 참여형 멀티미디어 응용의 개발을 가능하게 지원하는 미들웨어이다. 통신 계층은 분산 처리 환경의 메시지 전송을 담당하는 계층으로 MS-WINDOWS의 socket system을 활용한다. 즉, socket 함수의 family로는 AF_INET를, type은 비연결형 또는 연결형을 사용한다. 시스템 계층은 기본적으로 기존의 운영체제(예: Windows

95/98/2000/XP) 기능을 사용한다[16].



(그림 3) 두레의 구조

두레 계층은 상호 참여형 멀티미디어 일반적인 응용을 개발하기 위해서 설계된 프레임워크이다. 두레에서 제공되는 서비스 에이전트에는 여러 개의 기능들을 가진다. 이 에이전트들은 상호 협력 작업을 지원하기 위한 것으로서 세션 관리 에이전트, 접근/동시성 제어 에이전트, 오디오 혹은 미디어 자원의 공유를 가능하게 하는 미디어 제어 에이전트, 공동 작업 시 공동작업 공간(화이트보드 등)에서의 동일한 화면을 보게 하여 동시작업을 가능하게 하는 커풀링 에이전트, 전자우편 혹은 인터넷등 외부 네트워크와 접속을 담당하는 메일링 에이전트, 전체 세션에서 발생하는 세션의 종류, 이름, 참여자 명단, 통신의 양을 관리 하는 세션 감시 에이전트, 상용의 프리젠테이션 도구나 저작도구 등으로 개발된 소프트웨어를 공유하여 사용할 수 있게해 주는 응용공유 에이전트 등이 있

다. 또 이들의 외곽에는 통신 에이전트가 있어 여러 가지 통신 프로토콜을 지원 한다.

3.2 EDA

본 논문에서 제안하는 EDA는 여러 기능의 에이전트가 존재하며 원활한 오류 감지 및 복구 기법을 하는 Multi-agent 시스템이다. 이들 에이전트 간의 상호 협조를 통하여 비정상적으로 종료되어 inactive되어 있는 응용 S/W를 발견하고 이들 agent 간의 상호 협조를 통하여 자동적으로 active시켜주는 intelligent agent이다. EDA(Error Detection Agent)와 상호 연관이 되는 에이전트는 ECA(Error Classification Agent), ERA(Error Recovery Agent) 및 SMA(Session Management Agent) 이다. SMA는 다양한 서비스 객체들의 상호 작용에 의해서 지원된다. 최초 응용이 실행되면 응용은 데몬 객체의 존재를 찾아 응용으로써의 등록을 요구한다. 세션 생성을 위한 모든 필요한 준비가 끝나게 되면 세션 관리자는 응용 인터페이스를 통해 응용에게 세션이 생성되었음을 알려준다. 이후 응용은 세션에 대한 모든 서비스를 세션 관리자를 통해 요구할 수 있게 된다. EDA는 오류를 감지하는 핵심 agent로 fail detect 정보 흐름은 packet 정보를 보내어 그 응답 결과 상태를 분석하여 오류의 발생 여부를 분석한다. 세션의 복원을 진행하기 위해서는 먼저 에러 감지를 하기위한 방법이 필요하다. MS-Windows 98/2000/XP의 시스템에서 에러 감지를 위한 방법은 실행된 프로세서의 상태를 보관하는 프로세서 데이터 베이스를 주기적으로 검사하는 방법이다. 그러나 이것은 두레를 이용한 세션의 상태와는 무관한 프로세서

까지 데이터베이스를 검사해야 한다는 단점이 있다. 그래서 두레를 이용한 시스템에서는 데몬이나 세션 매니저가 생성한 프로세서에 대한 정보를 결합 감지기에게 통보하여 결합 허용 시스템에서 세션과 직접 연관된 프로세서만을 주기적으로 폴링(polling)하여 여러 감지를 수행한다. ECA는 사용자와 결합 감지 및 복구를 위한 EDA 및 ERA와 같이 에이전트를 기반으로 한 응용 분야와의 상호 작용을 위한 인터페이스로서의 역할을 하는 에이전트이다. ERA는 비정상적인 상황의 통보에 대하여 복원의 과정을 수행한다.

IV. 오류 감지 시스템의 모델링

분산 멀티미디어 환경 시스템 중에서 4번째 계층인 응용 계층과 3번째 계층인 두레 계층 사이의 관계만 국한 시킨다.

4.1 설계 요소(Design constraints)

오류 감지 및 복구 계층에서는 복구 기능은 제외시킨다. 즉, EDA에 대한 모델링을 통해서 본 시스템에 대한 범위를 한정한다. EDA에 대한 설명과 분석을 위해서 필요한 정의 및 표기는 다음과 같다.

(정의 1)

본 논문에 관계되는 Agent의 집합은 다음과 같다.
 $EDA = \{EDA_1, EDA_2, \dots, EDA_n\} (n \in \mathbb{N})$
 (EDA: 오류 감지 Agent 들의 집합)

(정의 2)

실제 환경 상태 P, 즉 오류 감지 또는 복구 대상

이 되는 미디어, 미디어 인스턴스 및 응용 프로그램들의 집합은 다음과 같다.

$$P = \{ p_1, p_2, \dots, p_n \} (n \in \mathbb{N})$$

P_i 는 t_i 와 t_j 사이의 시간 간격에서 실행하는 프로세스 들의 집합으로 정의한다.

$$\text{즉, } P_i = \{ p_i \mid t_i \leq p_i < t_j \} \text{ 이다.}$$

(정의 3)

E_i 는 t_i 와 t_j 사이의 시간 간격에서 발견되는 오류(error)들의 집합으로 정의한다.

$$E_i = \{ e_i \mid t_i \leq e_i < t_j \}$$

F_i 는 t_i 와 t_j 사이의 시간 간격에서 발생하는 오류의 원인이 되는 결함(fault)들의 집합으로 정의한다.

$$F_i = \{ f_i \mid t_i \leq f_i < t_j \}$$

S_i 는 t_i 와 t_j 사이의 시간 간격에서 SM이라는 세션 관리자를 이용하여 필요한 응용을 등록시키는데 발생하는 시간의 집합으로 정의한다.

$$S_i = \{ s_i \mid t_i \leq s_i < t_j \}$$

4.2 모델링(Modeling)

4.2.1 제안하는 알고리즘

제안하는 방식에 대한 알고리즘은 SM이라는 세션 관리자를 이용하여 등록되어 있는 필요한 응용만 찾아서 폴링(polling) 시간을 줄이는 방식이며 집합으로 표시하면 다음과 같다.

Set of Detection = {Set of error, Set of fault, Set of error detector}

여기에서

Set of error = {E, T, D}

(E: 발생하는 오류, T: 시간 간격,

D: 발생하는 도메인 위치)

Set of fault = {F, T, C}

(F: 발생하는 fault, T: 시간 간격,
C:PDB에서 찾은 error code
즉, GetExitCodeProcess 실행 후에
return되는 코드 값)

Set of error detector = {Addr_EDA,
Func_EDA}

(Addr_EDA: EDA의 주소 정보,

Func_EDA: EDA의 기능)

ti와 tj 사이의 시간 간격에 활동하는 EDAi의 기
능은 다음과 같다.

Function of EDAi = { pi x ei x si -> fi |
pi ∈ Pi, ei ∈ Ei, si ∈ Si, fi ∈ Fi}

오류 감지(error detect) 기능 중에서도 폴링 방
식에만 적용한다. 즉, 전체 응용에 대해서 폴링 방
식을 사용하여 오류를 감지하는 방식에만 적용하
며 SM이라는 세션 관리자를 이용하여 등록되어
있는 필요한 응용만 찾아서 폴링(polling)하는 방
식이다. PDB(Process Database)의 구조 중 프로
세스(process)의 살아있는 상태를 알려면 마침 상
태(termination status)의 내용을 보고 알 수 있다.
예를 들면, 하나의 프로세스가 여전히 활동하고
있으면 STILL_ACTIVE(검사 코드값: 103)이다.
PDB의 이 값을 알려면 API 함수 중
GetExitCodeProcess를 사용했을 때 반환(return)
되는 코드값으로 알 수 있다.

4.2.2 제안하는 방식과 기존 방식의 효율성 비교

본 논문에서는 전체 응용에 대해서 폴링 방식
을 사용하는 방식과 폴링을 사용하지 기존 시스템의
정보를 갖고 있는 SM이라는 세션 관리자를 이용
하여 등록되어 있는 필요한 응용만 찾아서 폴링
시간을 줄이는 방식 2가지를 비교하여 효율성 검
토를 할 수 있는 시스템의 모델링을 제안한다.

(1) 기존 방식

DEVS 형식론에서 원자 모델(atomic model)을 결
합하여 새로운 커플 모델(coupled model)을 형성
한다. 기존 방식에서 상태 변수는 표1과 같고
component interaction은 표2와 같다.

(표 1) 기존 방식에서 상태 변수

model	state variable	purpose
EF (genr)	poll_int	polling interval
RA1	ral_re_time app_count ral_re_t_a	response time 두레 관련 응용 갯수 반응시간 누적
UA1	ual_re_time app_count ual_re_t_a	response time 두레 무관 응용 갯수 반응시간 누적
ED1	ral_re_t_a ual_re_t_a tat1_t_a	ral 반응시간누적 ual 반응시간누적 ral+ual 누적

(표 2) 기존 방식에서 component interaction

순서	component interaction	message
step1	genr-> RA1 genr-> UA1	polling 발생 시점을 준다. RA1, UA1는 자신의 반응시 간을 계산한다.
step2	RA1-> ED1 UA1-> ED1	RA1, UA1는 자신의 ED1의 polling에 대한 반응시간을 준 다. ED1은 RA1, UA1로부터 의 반응시간을 누적한다. 간격 step1-step2 반복
step3	ED1-> EF1	ED1는 EF1에게 RA1, UA1 반응시간 누적 총합을 준다 (종료시점).

(2) 제안된 방식

DEVS 형식론에서 원자 모델(atomic model)을 결합하여 새로운 커플 모델(coupled model)을 형성한다. 제안된 방식에서 상태 변수의 정의는 표3과 같고 component interaction은 표4와 같다.

(표 3) 제안된 방식에서 상태 변수의 정의

model	state variable	purpose
EF(genr)	poll_int	polling interval
RA2	ra2_re_time app_count ra2_re_t_a	response time 응용 갯수 반응시간 누적
ED2	ra2_re_t_a sm_t_a tat2_t_a	ra2반응시간누적 SM 정보등록 시간 ra2시간누적 +SM정보등록시간

(표 4) 제안된 방식에서 component interaction

순서	component interaction	message
step1	genr → RA2	polling 발생 시점을 준다. RA2는 자신의 반응시간을 계산한다.
step2	RA2 → ED2	RA2는 자신의 ED2의 polling에 대한 반응시간을 준다. ED2는 RA2로부터의 반응시간을 누적한다. 간격 step1-step2 반복
step3	ED2 → EF2	ED2는 EF2에게 RA2 반응시간 누적과 SM정보등록시간 총합을 준다(종료시점).

(3) 시뮬레이션 결과

분산 멀티미디어 환경에서 실행되는 결합 허용 시스템의 기능 중에서 결합 감지 부분의 성능 분석을 DEVS 형식론을 이용하여 살펴 보았다. 즉,

simulation model을 통한 관측 목표와 관측 값 계산에 관련된 변수를 상태 변수로 가지는 모델이다. 오류 감지 부분에서 기존 방법의 감지 시간 (Tat1-t-a)은 다음과 같다.

$$Tat1-t-a = Poll_int * (App_count + App_count)$$

제안된 방법의 감지 시간(Tat2-t-a)은 다음과 같다.

$$Tat2-t-a = Poll_int * (App_count) + Sm_t_a$$

만일 프로세서 간의 메시지가 전달될 때 걸리는 시간을 t라고 하면 한번 폴링 시간은 2t가 된다. App_count > 1이므로 Poll_int * App_count > Sm_t_a이다. 즉, 제안된 방법이 더 효율적이다.

V. 결론

멀티미디어 통신의 발달로 분산 시스템의 중요성은 더해가고 있다. 분산 시스템은 하나의 노드 또는 자원이 결합이 발생하더라도 전체 시스템에 영향을 미치지 않기 때문에 결합 허용 시스템 설계를 위한 좋은 조건을 가진다. 수동적으로 전체 시스템의 망 관리를 하지 않고 자동적으로 결합을 감지하고 복구해야 하는 필요성은 증대되고 있다. 본 논문에서는 DEVS 형식론을 사용하여 실세계의 분산 멀티미디어 환경에서의 시스템을 모듈화된 가장 작은 단위로 genr, trnsducer, RA, UA, ED로 나누었다. genr과 trnsducer 각각의 atomic model을 결합하여 EF coupled model을 형성한다. 또한 UA, RA, ED를 결합하여 DP라는 coupled model을 형성하고 EF-DP를 결합하여 coupled model을 형성하는 것이다. 본 논문에서는 전체 응용에 대해서 폴링 방식을 사용하는 방식과 폴링을 사용하되 기존 시스템의 정보를 갖고 있는 SM이라는 세션 관리자를 이용하여 등록되어 있는 필요

한 응용만 찾아서 polling의 시간을 줄이는 방식 2가지를 비교하여 효율성 검토를 할 수 있는 시스템의 모델링을 제안하였다. 앞으로의 연구 방향은 이러한 시스템을 정형화하는 부분이 과제로 된다. 오류 감지 뿐만 아니라 오류 분류 및 복구 등에 대해서도 DEVS 형식론을 이용하여 분석하는 일이다. 또한 변수들의 값과 SM에 등록된 정보를 주는 시간을 통계화하고 일반화하여 모델링 및 시뮬레이션 하는 연구, 멀티미디어 분산에 관한 시뮬레이션에 관한 연구 등이 있다.

참고문헌

- [1] 박길철, 황대준, "멀티미디어 원격 교육 시스템 설계", 한국 정보 처리학회 멀티미디어시스템 연구회 학술대회 논문집, pp.54,1994.
- [2] Roy D. Pea, "Learning through multimedia", IEEE Computer Graphics & Application, Jul. 1991, pp.58-66.
- [3] Matthew E. Hodges, Russel M. Sasnett. Multimedia Computing-Case studies from MIT project Athena-.", Appison-Wtsley pub., 1993, pp.29-37.
- [4] Victoria Rosenborg, "A guide to multimedia", New Riders pub., 1993, pp.187-205.
- [5] 전준걸, 황대준, "상호 참여를 위한 탁상회의 시스템의 구현", 95년 한국 정보 과학지 가을 학술 발표 논문집 vol.22, No.2, pp.1041.
- [6] 박길철, 황대준, "네트워크 환경에서 멀티미디어 객체 동기화 모델 설계", 한국정보처리학회 제 1 권 2호, pp.568-571, 1994.10.
- [7] 장순주, 임종규, 정구영, 구용완, "분산 시스템에서 결합 허용성을 위한 프로세스 이주 연구", 한국 정보 과학회지 가을 학술발표 논문집 Vol.21, No2, pp. 132.1994.
- [8] 김영실, 백두권, 이산 사건 모델링과 시뮬레이션, 정보과학회지 제 13권 제 4호, 1995년 4월, pp.9-10.
- [9] Bernard P.Zeigler, "Object-Oriented Simulation with hierarchical, Modular Models", Academic Press, 1990.
- [10] Bernard P.Zeigler, "Multifaceted Modeling and Discrete Event Simulation, Orlando, FL: Academic, 1984.
- [11] Bernard P.Zeigler, "Theory of Modeling and Simulation, John iley, NY, USA, 1976, reissued by Krieger, Malabar, FL, USA, 1985.
- [12] 조대호, 김형중, 분산 전문가 시스템을 갖는 이산 사건 시뮬레이션: 제조 공정 오류 감지와 진단에의 적용, 한국 시뮬레이션학회 논문지 제7권 제 2호, 1998년 12월.
- [13] 지승도, 이종근, 이장세, 캠퍼스 네트워크의 구성 및 성능 분석 자동화 방법론, 한국 시뮬레이션학회 논문지 제7권 제 2호, 1998년 12월.
- [14] Chi, S.D., Lee, J.S., Lee, J.K. and Whang, J.H., NETE: Campuse Network Design Tool, in Proc. IASTED International Conference, July, 1997.
- [15] 김문희, "결합 허용 시스템의 설계 고려사항 및 동향", 정보과학회지, 제11권 제3호, pp.7, 1993.
- [16] Eung-Nam Ko, Dae-Joon Hwang, Implementation of a Fault-Tolerant System Running on DOORAE: FTSD, In proceedings of IEEE ISCE98, October 19-21, 1998, Taipei, Taiwan.



고 응 남

1984년 연세대 수학과(이학사)

1991년 숭실대 정보과학대학원
전산공학과 (공학석사)

2000년 성균관대 대학원 정보
공학과(공학박사)

1983년-1993년 대우통신 컴퓨터개발부 선임연구원

1993년-1997년 동우대학 전자계산과교수

1997년-2001년 신성대학 컴퓨터계열 교수

2001년-현재 천안대학교 정보통신학부 교수

관심분야 : 인터넷, 멀티미디어, CSCW, 결합허용,
에이전트 및 게임 등